



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/809,147	03/25/2004	Anson Horton	MS302712.1/MSFTP579US	7392
27195	7590	12/02/2008		
AMIN, TUROCY & CALVIN, LLP			EXAMINER	
127 Public Square			LEE, MARINA	
57th Floor, Key Tower				
CLEVELAND, OH 44114			ART UNIT	PAPER NUMBER
			2192	
			NOTIFICATION DATE	DELIVERY MODE
			12/02/2008	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

docket1@thepatentattorneys.com
hholmes@thepatentattorneys.com
lpasterchek@thepatentattorneys.com

Office Action Summary	Application No. 10/809,147	Applicant(s) HORTON ET AL.	
	Examiner MARINA LEE	Art Unit 2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 02 September 2008.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-6,8-23,26 and 27 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-6,8-23,26 and 27 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114.

Applicant's submission filed on September 02, 2008 has been entered.

2. This action is responsive to Amendment filed on September 02, 2008. Claims 1, 2, 8, 9, 11, 13, 21, 26, and 27 have been amended. Claims 7, 24, and 25 have been cancelled. No claims have been newly added. Thus, Claims 1-6, 8-23, and 26 are presented for examination.

Prior Art's Arguments - Rejections

3. Applicant's arguments filed on September 02, 2008, especially on pages 7-11 of Remarks, with respect to new claim limitation, "...values of one or more properties of... being debugged...the value of the one or more properties ... implemented as a private nested ... value of the one or more properties of the object to a developer.", currently recites in claims 1, 21, 26, and 27 respectively , have been fully considered but they are moot in view of new ground rejection under (Dandoy (US 2004/0230954 A1 of record) alone or in combination of Bates (U.S. Patent No. 6,961,924 B2 of record) as will be further addressed bellow.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

Art Unit: 2192

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 1-6, 8-18 and 27 are rejected under 35U.S.C. 103(a) as being unpatentable over Dandoy, (U.S. Patent No. 2004/0230954 A1- art of record).

As to claim 1, Dandoy discloses a computer-implemented attributed debugging system (e.g., system of Fig. 1 – see at least [0017]) comprising:

a debugger (e.g., UI Debugger such as 110 or 200 of Figs 1 and 2) that facilitates debugging of a computer software application (e.g., application 100 (fig.1)) – See at least [0017], [0018], and [0029];

an expression evaluator (e.g., Debug Agent such as 115, 205, or 300 of Figs 1, 2, or 3 respectively) associated with the debugger that determines values of one or more properties of an object of the computer software application being debugged, the expression evaluator determines the values of the one or more propitiates of the object based upon a display proxy (*e.g., run time objects of application 100—See at least [0022]*) the display proxy provides relevant features of the object and conceal implementation specifics of the object; --(*e.g., the debug agent 115 configures to collect execution data relating to the graphical user interface 105 during run-time such as object properties, events associated with objects, run-time states of the application 100, and/or other desired run-time data – see at least [0018] - with emphasis added*), the display proxy provides relevant features (*e.g., the debug agent 115 can include logic configured to change properties of objects that are generated by software application 100*

Art Unit: 2192

during run-time in response to user wishes such as hide (conceal) visibility of certain objects within the level objects or within a child/parent objects (i.e., debug agent 300) -- see at least [0025], [0026], and [0041] with emphasis added ; and

a variable display component that presents the determined details of the object to a developer (e.g., object identity such name/type or hierarchy of objects can be determined and displayed upon user's requests – see at least [0046]).

It noted that Dandoy does not explicitly disclose a display proxy implemented as a private nested class of the object. However, Dandoy teaches e.g., the debug agent 115 can include logic configured to change properties of objects that are generated by software application 100 during run-time in response to user wishes such as hide (conceal) object, show an object, change color or other properties.. and it is further to note that the debug agent 115 can be configured to use what is called 'reflection' which is the ability to see/access at runtime names of objects and their method/properties by using get method (i.e. getX()). – See at least [0025] and [0026] – with emphasis added.

Thus, it would have been obvious to one ordinary skill in the art at the time invention was made to realize that the private class of the objects application 100 must have been implemented some how for the debug agent 115 to gain accessed to the hidden properties by using the known method (get X ()) during 'reflection' object configuration as taught in (e.g., [0026]).

As to claim 2, Dandoy further discloses the expression evaluator evaluates an expression associated with the object to determine the values (e.g., the debug agent 115 can be configured to use what is called 'reflection' which is

Art Unit: 2192

the abilities to see at runtime names of objects and their methods – see at least [0026] - with emphasis added), the expression is implemented in an particular programming language (e.g., *debugger implementation in various programming languages and tools such as JAVA, HTML, Java Server Pages (JSP), Pascal, C#, C++, C, CGI, Perl, APIs, SDKs, assembly, firmware, microcode, and/or other languages and tools – see at least [0048] and [0049] -- with emphasis added*).

As to claim 3, Dandoy further discloses the programming language comprising at least one of C#, J# or Visual Basis.Net (e.g., *debugger implementation in various programming languages and tools such as JAVA, HTML, Java Server Pages (JSP), Pascal, C#, C++, C, CGI, Perl, APIs, SDKs, assembly, firmware, microcode, and/or other languages and tools – see at least [0048] and [0049] -- with emphasis added*).

As to claim 4, Dandoy also discloses further comprising a plurality of expression evaluators, wherein each expression evaluator is associated with a different programming language (e.g., *the various component of debugger (e.g., debugger agent) can be implement in various programming languages and tools such as JAVA, HTML, Java Server Pages (JSP), Pascal, C#, C++, C, CGI, Perl, APIs, SDKs, assembly, firmware, microcode, and/or other languages and tools – see at least [0048] and [0049] -- with emphasis added*)).

As to claim 5, Dandoy also discloses the object comprises a class that includes at least one of a property or method (e.g., Object properties may include, for example, class, position, size, visibilities, and other properties – see at least [0041]).

As to claim 6, Dandoy further discloses the expression evaluator creates an instance of the display proxy associated (e.g., object/image on user interface 105) with the object of the computer software application (e.g., *the debug agent 115 configures to collect execution data relating to the graphical user interface 105 during run-time such as object properties, events associated with objects, run-time states of the application 100, and/or other desired run-time data – see at least [0018] - with emphasis added*).

As to claim 8, Dandoy further discloses the display proxy has access to private implementation specifics of the object (e.g., *using get x() method to emit the name of the object at runtime – see at least [0026] – with emphasis added*).

As to claim 9, Dandoy discloses further comprising an attribute cache directory that stores an attribute associated with the display proxy, the expression evaluator employs the stored attribute to determine the details of the object (e.g., *the debug agent 115 is configured to collect execution data (i.e. object properties...) relating to the graphical user interface 105 during run-time, where the execution data can be obtained via reading state values stored in memory that are maintained by the application during execution – see at least [0018] – with emphasis added*).

As to claim 10, Dandoy discloses further comprising an editing component that facilitated modifying a value associated with the object (e.g., *the debug agent 115 can be configures to call a method in the application 100 to change/edit properties of objects such as visibility, colors or other property*

Art Unit: 2192

changes (i.e., hierarchy (parent/child)) – see at least [0025],[0026], and [0041-- with emphasis added).

As to claim 11, Dandoy also discloses the variable display employs at least one attribute associated with the object that provides a format to display the determined values of one or more properties of the object (*e.g., the debug agent 115 can be configured to display/show properties of objects according to changing of certain colors or other properties (i.e., hierarchy (parent/child)) – see at least [0025],[0026], and [0041-- with emphasis added).*

As to claim 12, Dandoy discloses the attribute specifies at least one of whether a property of the object is displayed or how the property is displays (*e.g., the debug agent 115 can be configured to display/show properties of objects according to certain colors or other properties (i.e., hierarchy (parent/child)) – see at least [0025], [0026], and [0041-- with emphasis added).*

As to claim 13, Dandoy also discloses the attribute employs an enumeration to specify the format of the display (*e.g., debug agent is configured to collect execution data relating to the graphical user interface, which includes object properties, events, and runtime states for displaying/showing properties of objects according to certain colors or other properties (i.e., hierarchy (parent/child)) – see at least [0018],[0025], [0026], and [0041]-- with emphasis added).*

As to claim 14, Dandoy further discloses one enumeration value that indicates the property should not be displayed to the developer (*e.g., the debug*

Art Unit: 2192

agent can determine the current state of the selected window and it changes its properties: hidden see at least [0025]).

As to claim 15, Dandoy further discloses the enumeration includes one enumeration value that indicates a hierarchical property is expanded by default (e.g., at any points during debugging, a hierarchy objects within the interface can be determined and displayed, the hierarchy can be displayed automatically see [0046]).

As to claim 16, Dandoy also discloses the enumeration includes one enumeration value that indicates a hierarchical property is not expanded by default (e.g., debugging requests may include a request to monitor events associated with an object ... or request to hide or show and object see [0024]).

As to claim 17, Dandoy further discloses the enumeration includes one enumeration value that indicates a hierarchical property itself is not displayed and members of the hierarchical property are display (e.g., at any points during debugging, a hierarchy objects within the interface can be determined and displayed – see [0046]).

As to claim 18, Dandoy also discloses the attribute specifies what is displayed for a class (*e.g., the debug agent 115 can be configured to display/show properties of objects according to changing of certain colors or other properties (i.e., hierarchy (parent/child) within user request) – see at least [0025],[0026], and [0041]-- with emphasis added*).

As to claim 27, Dandoy discloses a computer-implemented attributed debugging system (e.g., system of Fig. 1 – see at least [0017]) comprising:

a debugger (e.g., UI Debugger such as 110 or 200 of Figs 1 and 2) that facilitates debugging of a computer software application (e.g., executable application 100 (fig.1)) – See at least [0017], [0018], and [0029];

an expression evaluator (e.g., Debug Agent such as 115, 205, or 300 of Figs 1, 2, or 3 respectively) associated with the debugger that examines at least one of the computer software application, the expression evaluator determines debug information comprising states of the at least one object, the state include values for at least one properties of the at least one object (e.g., *the debug agent 115 configures to collect execution data relating to the graphical user interface 105 during run-time such as object properties, events associated with objects, run-time states of the application 100, and/or other desired run-time data – see at least [0018] - with emphasis added*).

the expression evaluator inspects the at least one object to verify if each of the at least one object includes a display proxy ((e.g., *run time objects of application 100—See at least [0022]*), the display proxy provides relevant properties of an associated object and conceal properties related to implementation of the object, the expression evaluator examines the display proxy to determine debug information that includes for the relevant properties of the at least one object(e.g., *the debug agent 115 can include logic configured to change properties of objects that are generated by software application 100 during run-time in response to user wishes hide (conceal) visibility of certain objects within the level objects or within a child/parent objects (i.e., debug agent 300) -- see at least [0025], [0026], and[0041] with emphasis added* ;

an attribute cache directory that retains instances of one more display proxies, the expression evaluator queries the attribute caches directory for instance of display proxies associated with the at least one object, the expression evaluator creates an instance if not retained in the attribute cache directory – *(e.g., the debug agent 115 is configured to collect execution data (i.e. object properties...) relating to the graphical user interface 105 during run-time, where the execution data can be obtained via reading state values stored in memory that are maintained by the application during execution (cache)– see at least [0018] – with emphasis added);*

a variable display component that presents the debug information to a developer, the debug information includes values of relevant properties of the at least one object determined from examination of display proxies of the at least one object (e.g., *object identity such name/type or hierarchy of objects can be determined and displayed upon user requests – see at least [0046]*).

It noted that Dandoy does not explicitly disclose a display proxy implemented as a private nested class of the object. However, Dandoy teaches *e.g., the debug agent 115 can include logic configured to change properties of objects that are generated by software application 100 during run-time in response to user wishes such as hide (conceal) object, show an object, change color or other properties.. and it is further to note that the debug agent 115 can be configured to use what is called 'reflection' which is the ability to see/access at runtime names of objects and their method/properties by using get method (i.e. getX()). – See at least [0025] and [0026] – with emphasis added.*

Thus, it would have been obvious to one ordinary skill in the art at the time invention was made to realize that the private class of the objects application 100 must have been implemented some how for the debug agent 115 to gain access to the hidden properties by using the known method (get X ()) during 'reflection' object configuration as taught in (e.g., [0026]).

6. Claims 19 and 20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dandoy, (U.S. Patent No. 2004/0230954 A1) in view of Bates et al., (U.S. Patent No. 6,961,924 B2 – hereinafter Bates)

As to claim 19, it is noted that Dandoy does not explicitly disclose the attributes includes an argument that comprises a string that is displayed in a value column for an instance of the class. However, Bates, in an analogous art teaches, the debugger 123 determines whether any attributes are set for the variable. If any attributes you set for the variable, then processing to step 616 where the appropriate attribute indicator (e.g., G, S, I, R, C, P) for each set attribute is associated with the variable value and determines whether the variable value is associated with a fields of a record (class) in a symbol table 120, that is the debugger determines whether a field is to be displayed(see step 614 and 618, Fig. 6, col. 11: 32-50; also see attributes are set in the fields 312-322, Fig. 3 and associated text, and step 616, col. 11: 34-37 – with emphasis added).

It would have been obvious to one ordinary skill in the art at the time invention was made to use the attribute indicator of the debugger 123 of Bate

Art Unit: 2192

with debug agent of Dandoy as convenient way of accessing run time variable value to user as taught in step 616, col. 11: 34-37 of Bates.

As to claim 20, Bates discloses the argument is associated with a property of the class (e.g., at step 618, the debugger 123 determines whether the variable value is associated with a field of a record (class) –see col. 11: 41-42).

7. Claim 21-23 and 26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bates et al., (U.S. Patent No. 6,961,924 B2) in view of Dandoy, (U.S. Patent No. 2004/0230954 A1)

As per claims 21 and 26, Bates discloses a method facilitating attributed debugging comprising:

receiving a request to examine an object in a computer software application being debugged(e.g., an event, at step 604, may be in response to user...entering a request for a variable value in a command line, etc. see col. 11: 16-22);

determining whether a display proxy attribute exists for the object (e.g., at step the debugger 123 determines whether any attributes are set for the variable see col. 11: 32-33);

creating a display proxy for the object in accordance with the display proxy attribute; and (e.g., the attributes are set in the fields 312-322 see col. 11: 33-34) ; and

examining the display proxy to determine debug information(see step 616 and 618 col. 11: 34-50).

It is noted that Bates does not explicitly disclose the display proxy is implemented as a private nested class of the object such that the display proxy within the definition of the object, the display proxy provides relevant properties regarding a state of the object and conceals properties related to implementing of the object. However, Dandoy, in an analogous art, teaches how debug agent (*i.e.*, debug agent 115) include logic configured to change properties of objects that are generated by software application during run-time in response to user wishes such as hide (conceal)/show (display) visibility of certain objects within the level objects or within a child/parent objects (*i.e.*, debug agent 300) -- see at least [0025], [0026], and [0041] with emphasis added.

It would have been obvious to one ordinary skill in the art at the time invention was made to modify the debugger 123 of Bates with the debugging collecting agent with visibilities setting (e.g., show/hide certain object/child) of Dandoy for saving programmer time in debugging system with more efficient way of debugging without modifying the source code as once supplied by Dandoy as in page 1, [0001] and [0002] with emphasis added.

Further regarding to claim 26, Bates discloses an attributes debugging system (e.g., computer 110, Fig. 1 see col. 4: 28-40 and related text) comprising: means for implementing the method regarding to claim 21.

As to claim 22, Bates discloses further comprising providing the debug information determined by the display proxy to the developer in response to the request to examine the object (e.g., the debugger 123 determines whether any attributes are set for the variable and determines whether the variable value is

Art Unit: 2192

associated with a fields of a record, that is the debugger determines whether a field is to be displays see step 614 and 618, Fig. 6, col. 11: 32-50).

As to claim 23, Bates discloses a computer readable medium (e.g., CD-ROM disk see col. 3: 6-12) having stored thereon computer executable instructions for carrying out the method of claim 21.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to the applicant disclosure.

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Marina Lee whose telephone number is (571) 270-1648. The examiner can normally be reached on M-F (11am-7: 30pm) Est.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service

Art Unit: 2192

Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/M. L./
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192